

What is Claimed is:

1. A method for supporting development of content independent of a run-time platform, comprising the steps of:
 1. storing processing blocks that define content; and
 2. storing an application graph that expresses the identity of the stored processing blocks and data connectivity between the stored processing blocks; whereby, the application graph can be traversed by a graphical application platform at run-time to execute appropriate processing blocks on a run-time platform.
2. The method of claim 1, wherein the content comprises game content.
3. A method for supporting development of content independent of multiple hardware platforms, comprising the steps of:
 1. storing processing blocks that define content independent of multiple hardware platforms;
 2. selecting a target hardware platform from multiple hardware platforms;
 3. storing an application graph that expresses the identity of the stored processing blocks and data connectivity between the stored processing blocks based on the selected target hardware platform; and
 4. traversing the application graph at run-time, including executing appropriate processing blocks on the selected target hardware platform.
4. The method of claim 3, wherein the content comprises game content, and the multiple hardware platforms include at least one of a game console platform and a personal computer platform.
5. A game development and run-time system, comprising:

a graphical application platform that enables a game application to run on any of multiple hardware platforms.

5 6. The system of claim 5, further comprising:

an object definition tool that enables a developer to define an application graph such that said game application can run on a target hardware platform.

10 7. The system of claim 6, wherein said object definition tool further enables a developer to define objects, object elements, and connections.

15 8. A graphical application platform for leveraging capabilities provided independently in at least one of an application software and a hardware platform, comprising:

an application real-time kernel (ARK);
a plurality of standard features implemented as executable blocks of logic; and

20 connections between said blocks that implement data flow between said blocks, whereby capabilities of at least one of the application software and the hardware platform can be implemented modularly by adding additional corresponding blocks and connections.

25 9. The graphical application platform of claim 8, wherein said ARK comprises logic that invokes blocks according to a schedule listing the blocks to be executed in each of at least one ARK thread running on at least one central processing unit, dynamically loads and unloads blocks, monitors block execution, and facilitates thread management, memory sharing, mutual exclusion, and synchronization.

10. The graphical application platform of claim 8, wherein said additional blocks implement additional features, said additional features comprising market oriented features.

5 11. The graphical application platform of claim 8, wherein said additional blocks implement additional features, said additional features comprising application specific features.

10 12. The graphical application platform of claim 8, wherein said standard and additional blocks are organized into components, wherein each component comprises blocks representing alternative implementations of a feature.

15 13. The graphical application platform of claim 12, wherein each of said alternative implementations comprises:

- a) blocks corresponding to said alternative implementation;
- b) identification of resources needed by said alternative implementation; and
- c) identification of resources provided by said alternative implementation.

20 14. A method of pre-processing a graphics application with respect to a predefined hardware platform, comprising the steps of:

- a) selecting from among a set of alternative implementations of a feature;
- b) mapping at least one block, corresponding to the selected implementation, to a phase of execution;
- c) mapping the phase of execution to a stage of execution;
- d) creating a block execution order list corresponding to the stage of execution; and

e) submitting the stage of execution to an application real time kernel for management of execution of the stage.

15. The method of claim 14, wherein said step a) comprises a negotiation process in which resource requirements of each alternative implementation are considered, along with the costs and benefits of variations in such resource requirements, thereby allowing selection of an implementation.

5